

Date=16/12/2020

Lecture By= Akash Handa

Subject ⇒ Create Login Form in React

IN PREVIOUS LECTURE (QUICK RECAP) Date-15/12/2020	In Today's Lecture (Overview)
<p>Forms in React</p> <ul style="list-style-type: none">Controlled ComponentsHandling Multiple Inputs <p>JSON Server</p> <ul style="list-style-type: none">Getting started	<p>Login App – Create login form in ReactJS using secure REST API</p> <p>Way to create login form in ReactJS using secure REST API</p> <ol style="list-style-type: none">1. Create secure REST API2. Setup react application3. Create react components like Home, Login and Dashboard

Login App – Create login form in ReactJS using secure REST API

Way to create login form in ReactJS using secure REST API

1. Create secure REST API
2. Setup react application
3. Create react components like Home, Login and Dashboard
4. Implement authenticated routes
5. Output

1. Create secure REST API

To create login application, we need secure REST API to integrate into the application. So we have already created the REST API in Node.js for authentication. Kindly refer the below article to create it or clone it from the GitHub repository.

[Login App – Create REST API for authentication in Node.js using JWT – Part 2](#)

After that just run the project so we can consume the REST API.

2. Setup react application

First, Let's setup the simple react application to implement the login functionality. Following link will help you to create basic react application.

Create React Application

3. Create react components like Home, Login and Dashboard

Now, It's time to create components for Home, Login and Dashboard pages and link it using the routing.

- Home component (Access it with/without login) – It's a simple component which we can access it with/without user login. By default we'll open this component.
- Login component (Access it without login only) – Here we will create a login page. So with the help of it we can call the REST API and on successful call we will redirect it on the Dashboard component.
- Dashboard component (Access it after login only) – It's a private page that users can access only after successfully logged in. We will also provide the option to logout from user account in the same component.

If you don't know how to implement routing in react app then refer to the following link for routing.

[“Routing in React JS”](#)

For demo purpose we are creating these components and updating the css in “src” directory.

Home.js

```
import React from 'react';

function Home() {
  return (
    <div>
      Welcome to the Home Page!
    </div>
  );
}
```

```
export default Home;
```

Login.js

```
import React, { useState } from 'react';

function Login(props) {
  const username = useFormInput('');
  const password = useFormInput('');
  const [error, setError] = useState(null);
  const [loading, setLoading] = useState(false);

  // handle button click of login form
  const handleLogin = () => {
    props.history.push('/dashboard');
  }

  return (
    <div>
      Login<br /><br />
      <div>
        Username<br />
        <input type="text" {...username} autoComplete="new-password" />
      </div>
      <div style={{ marginTop: 10 }}>
        Password<br />
        <input type="password" {...password} autoComplete="new-password" />
      </div>
      {error && <><small style={{ color: 'red' }}>{error}</small><br /></></><br />
      <input type="button" value={loading ? 'Loading...' : 'Login'}
        onClick={handleLogin} disabled={loading} /><br />
    </div>
  );
}

const useFormInput = initialValue => {
  const [value, setValue] = useState(initialValue);
```

```

    const handleChange = e => {
      setValue(e.target.value);
    }
    return {
      value,
      onChange: handleChange
    }
  }
}

export default Login;

```

Dashboard.js

```

import React from 'react';

function Dashboard(props) {

  // handle click event of logout button
  const handleLogout = () => {
    props.history.push('/login');
  }

  return (
    <div>
      Welcome User!<br /><br />
      <input type="button" onClick={handleLogout} value="Logout" />
    </div>
  );
}

export default Dashboard;

```

App.js

```

import React from 'react';
import { BrowserRouter, Switch, Route, NavLink } from 'react-router-dom';

import Login from './Login';
import Dashboard from './Dashboard';
import Home from './Home';

function App() {

```

```

return (
  <div className="App">
    <BrowserRouter>
      <div>
        <div className="header">
          <NavLink exact activeClassName="active" to="/">Home</NavLink>
          <NavLink activeClassName="active"
to="/login">Login</NavLink><small>(Access without token only)</small>
          <NavLink activeClassName="active"
to="/dashboard">Dashboard</NavLink><small>(Access with token only)</small>
        </div>
        <div className="content">
          <Switch>
            <Route exact path="/" component={Home} />
            <Route path="/login" component={Login} />
            <Route path="/dashboard" component={Dashboard} />
          </Switch>
        </div>
      </div>
    </BrowserRouter>
  </div>
);
}

export default App;

```

Index.js

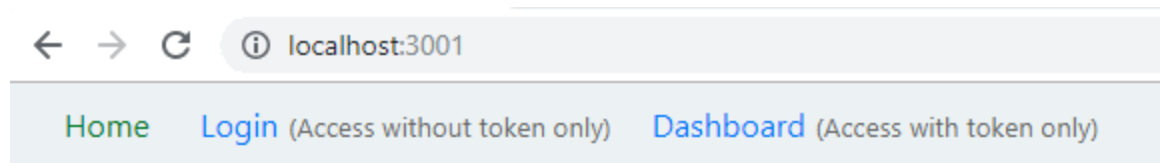
```

export default App;
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));

```

Output



Github for the Above Code

<https://github.com/cluemediator/login-app-reactjs>

For more like this Click on the link below

<https://www.educba.com/react-login-form/>